

Data Collection and Preprocessing Phase

Date	25 September 2024
Team ID	SWTID1726888137
Project Title	intelligent handwritten digit identification system for computer applications
Maximum Marks	6 Marks

Preprocessing Template

The images will be preprocessed by resizing, normalizing, augmenting, denoising, adjusting contrast, detecting edges, converting color space, cropping, batch normalizing, and whitening data. These steps will enhance data quality, promote model generalization, and improve convergence during neural network training, ensuring robust and efficient performance across various computer vision tasks.

Section	Description
Data Overview	the MNIST, which contains 70,000 grayscale images of digits (0–9) with a 28x28 pixel resolution, used for training and testing the model.
Resizing	Standardizes all images to a fixed size (e.g., 28x28) to ensure consistency and reduce computation.
Normalization	Scales pixel values from 0 to 1, improving model stability and training speed.
Data Augmentation	Expands the dataset by creating varied copies (e.g., rotated, shifted, zoomed) to help the model generalize to new data.
Color Space Conversion	Converts images to grayscale if needed, simplifying data and reducing computational load for consistent input.
Image Cropping	Removes unwanted borders or whitespace around the digit, focusing on the main features to improve model accuracy and efficiency.

Batch Normalization	Normalizes layer outputs within each mini-batch, speeding up training and enhancing model stability.
Data Preprocessing Code Screenshots	
Loading Data	<pre># Load dataset directly from keras library (X_train, y_train), (X_test, y_test) = mnist.load_data()</pre>
Resizing	<pre># Resize ROI image to 28x28 pixels img = cv2.resize(roi, (28, 28), interpolation=cv2.INTER_AREA)</pre>
Normalization	<pre># Normalize inputs X_train = X_train / 255 X_test = X_test / 255 # Normalize the image to support model input img = img / 255.0</pre>
Data Augmentation	<pre># Data Augmentation datagen = ImageDataGenerator(... rotation_range=10, ... width_shift_range=0.1, ... height_shift_range=0.1, ... zoom_range=0.1, ... horizontal_flip=False) datagen.fit(X_train)</pre>
Color Space Conversion	<pre># Convert the image to grayscale gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)</pre>
Image Cropping	<pre>roi = th[y - top:y + h + bottom, x - left:x + w + right]</pre>
Batch Normalization	<pre>model.add(BatchNormalization())</pre>